CHAPTER 4

System Management

Introduction

This chapter provides information about operation and management tasks for the Cisco H.323 Signaling Interface (HSI) application. This chapter contains the following sections:

Restarting the Cisco HSI Application

To restart the Cisco HSI at the MML command prompt, use the **restart-softw** MML command. For more information about this command, see Appendix A, "MML Commands."

To start the Cisco HSI application, see the "Starting the Cisco HSI" section on page 2-10.

Stopping Call Processing

To stop call processing, use the **stp-callproc** MML command. This command causes the handling of new call requests to cease immediately, and, if no timeout period is specified, all existing calls are released immediately. If a timeout period is specified, existing calls are released after the specified amount of time has elapsed. For more information about the **stp-callproc** command, see Appendix A, "MML Commands."

Starting Call Processing

To start call processing, use the **sta-callproc** MML command. For more information about this command, see Appendix A, "MML Commands."

Stopping the Call Processing Application

To stop the call processing application, use the **stp-softw** MML command. For more information about this command, see Appendix A, "MML Commands."

Starting the Call Processing Application

To start the call processing application, use the **sta-softw** MML command. For more information about this command, see Appendix A, "MML Commands."

Reporting the Cisco HSI Status

To display the status of the Cisco HSI, use the **rtrv-softw** MML command. For more information about this command, see Appendix A, "MML Commands."

Measurements

The following sections describe the two measurement categories:

- System-related measurements
- Call-related measurements

System-Related Measurements

The CIagent is a Simple Network Management Protocol (SNMP) subagent. It handles the collection and storage of the following system performance measurements:

- CPU occupancy
- RAM occupancy
- Disk occupancy
- TCP usage

Use the CIAGENTSCANPERIOD parameter to define the period that the CIagent polls the CPU utilization (see Chapter 5, "Provisioning the Cisco HSI").

Call-Related Measurements

The Cisco HSI application handles all call-related measurements. An SNMP MIB handles the collection of call-related measurement data.

The call-related measurements are organized into counter groups. The following MML counter groups are required:

- RAS (see Table 4-1 on page 4-3)
- Q.931 (see Table 4-2 on page 4-4)
- H.245 (see Table 4-3 on page 4-5)

The measurements in these groups are written to disk every 30 minutes in a file. The file name includes the date and time that measurements were written to disk.

Table 4-1 RAS Counter Group

Counter Name	Measurement	Type	Comments
GK_DISC_ATT_TOT	Gatekeeper Discovery Attempts	Integer	Incremented for every unicast gatekeeper request (GRQ) sent or for every multicast operation
GK_REG_ATT_TOT	Registration Request Attempts	Integer	Incremented for every registration request (RRQ) sent
GK_REG_SUCC_TOT	Registration Request Successes	Integer	Incremented for every registration confirmation (RCF) received
GK_RCV_UNR_ATT_TOT	GK Initiated Unregistration Attempts	Integer	Incremented for every unregistration request (URQ) received from a gatekeeper (GK)
GK_XMIT_UNR_SUCC_TOT	GK Initiated Unregistration Successes	Integer	Incremented for every unregistration confirmation (UCF) sent to a GK
GK_XMIT_UNR_ATT_TOT	TC Initiated Unregistration Attempts	Integer	Incremented for every URQ sent to a GK
GK_RCV_UNR_SUCC_TOT	TC Initiated Unregistration Successes	Integer	Incremented for every UCF received from a GK
GK_RLS_ATT_TOT	Disengage Attempts	Integer	Incremented for every disengage request (DRQ) sent to a GK
GK_RLS_SUCC_TOT	Disengage Successes	Integer	Incremented for every disengage confirmation (DCF) returned by a GK
GK_INFO_REPORT_TOT	Information Reports	Integer	Incremented for every information request (IRQ) sent to the GK

Table 4-2 Q.931 Counter Group

Counter Name	Measurement	Type	Comments
FC_INC_CALL_ATT_TOT	H.225 Incoming Fast Connect Call Attempts	Integer	Incremented when a setup containing the fastStart element is received
FC_INC_CALL_SUCC_TOT	H.225 Incoming Fast Connect Call Successes	Integer	Incremented when the Fast Connect procedure is used to establish an incoming H.323 call
FC_OTG _CALL_ATT_TOT	H.225 Outgoing Fast Connect Call Attempts	Integer	Incremented when a setup containing the fastStart element is sent to an H.323 endpoint Decremented when reverting to Version 1 signaling (and another measurement incremented)
FC_OTG_CALL_SUCC_TOT	H.225 Outgoing Fast Connect Call Successes	Integer	Incremented when the Fast Connect procedure is used to establish an outgoing H.323 call
V1_INC_CALL_ATT_TOT	H.225 Incoming Version 1 Call Attempts	Integer	Incremented when an incoming H.323 Version 1 Setup is received (that is, no fastStart element or H.245 tunneling)
V1_INC_CALL_SUCC_TOT	H.225 Incoming Version 1 Call Successes	Integer	Incremented when an incoming H.323 Version 1 call is established
V1_OTG_CALL_ATT_TOT	H.225 Outgoing Version 1 Call Attempts	Integer	Incremented when an outgoing H.323 call reverts to Version 1 signaling
V1_OTG_CALL_SUCC_TOT	H.225 Outgoing Version 1 Call Successes	Integer	Incremented when an outgoing H.323 call using Version 1 is established
INC_NORM_REL_TOT	H.225 Incoming Call Normal Releases	Integer	Incremented when an established incoming H.323 call is taken down due to user on-hook
INC_ABNORM_REL_TOT	H.225 Incoming Call Abnormal Releases	Integer	Incremented when an established incoming H.323 call is taken down due to anything other than user on-hook
OTG_NORM_REL_TOT	H.225 Outgoing Call Normal Releases	Integer	Incremented when an established outgoing H.323 call is taken down due to user on-hook
OTG_ABNORM_REL_TOT	H.225 Outgoing Call Abnormal Releases	Integer	Incremented when an established outgoing H.323 call is taken down due to anything other than user on-hook

Table 4-3 H.245 Counter Group

Counter Name	Measurement	Type	Comments
MASTER_SLAVE_ATT_TOT	H.245 Master Slave Determination Attempts	Integer	Incremented whenever either side of the call initiates the Master Slave Determination procedure (by either H.245 tunneling or a separate H.245 signaling path)
MASTER_SLAVE_SUCC_TOT	H.245 Master Slave Determination Successes	Integer	Incremented whenever a Master Slave Determination procedure is completed
TERM_CAP_XCHG_ATT_TOT	H.245 Terminal Capability Exchange Attempts	Integer	Incremented whenever either side of the call initiates the Capability Exchange procedure (by either H.245 tunneling or a separate H.245 signaling path)
TERM_CAP_XCHG_SUCC_TOT	H.245 Terminal Capability Exchange Successes	Integer	Incremented whenever a Capability Exchange procedure is completed
OPEN_CH_ATT_TOT	H.245 Open Logical Channel Attempts	Integer	Incremented whenever either side of the call initiates the Open Logical Channel procedure (by either H.245 tunneling or a separate H.245 signaling path)
OPEN_CH_SUCC_TOT	H.245 Open Logical Channel Successes	Integer	Incremented whenever an Open Logical Channel procedure is completed
CLOSE_CH_ATT_TOT	H.245 Close Logical Channel Attempts	Integer	Incremented whenever either side of the call initiates the Close Logical Channel procedure (by either H.245 tunneling or a separate H.245 signaling path)
CLOSE_CH_SUCC_TOT	H.245 Close Logical Channel Successes	Integer	Incremented whenever a Close Logical Channel procedure is completed
AVG_ROUND_TRIP_DELAY	H.245 Round Trip Delay Determination	Average (ms)	The average time (in ms) for Round Trip Delay measured as a result of successful Round Trip Delay Determination procedures

Resetting Measurements

The **clr-meas** MML command resets the measurement counters. This command resets an individual counter or all counters in a counter group. The following are valid counter groups:

- RAS
- Q.931
- H.245

For more information about the **clr-meas** command, see Appendix A, "MML Commands."

Retrieving Counters

Use the **rtrv-ctr** MML command to retrieve measurement counters. This command displays the measurements for a counter group. Valid counter groups are RAS, Q.931, and H.245. For more information about the **rtrv-ctr** command, see Appendix A, "MML Commands."

Overload

The system continuously checks call total and CPU utilization. Each of these values is compared to predefined limits. For the call total, three limits are available. Each limit has a hysteresis value and an alarm associated with it. When the call total reaches the limit, an alarm is raised. When the call total falls below the limit minus the hysteresis value, the alarm is cleared when the appropriate recovery action is taken.

Cisco HSI supports the ability to maintain the following three levels of overload:

- Overload level 1
- Overload level 2
- Overload level 3

The following factors can trigger each level of overload:

- CPU usage (the OVLDSAMPLERATE parameter defines the frequency of CPU sampling and threshold checking)
- Maximum calls allowed

Disk usage can trigger a LOW_DISK_SPACE alarm. For more information about this alarm, see Chapter 6, "Cisco HSI Alarms and Troubleshooting."

See Chapter 6, "Cisco HSI Alarms and Troubleshooting," for information about overload alarms.

Overload Level 1

Use the following configuration parameters for overload level 1 (see Chapter 5, "Provisioning the Cisco HSI"):

- OVLDLEVEL1PERCENT
- OVLDLEVEL1FILTER
- OVLDLEVEL1THRESHLOWERCALLS
- OVLDLEVEL1THRESHUPPERCALLS
- OVLDLEVEL1THRESHLOWERCPU
- OVLDLEVEL1THRESHUPPERCPU

Overload Level 2

Use the following configuration parameters for overload level 2 (see Chapter 5, "Provisioning the Cisco HSI"):

- OVLDLEVEL2PERCENT
- OVLDLEVEL2FILTER
- OVLDLEVEL2THRESHLOWERCALLS
- OVLDLEVEL2THRESHUPPERCALLS
- OVLDLEVEL2THRESHLOWERCPU
- OVLDLEVEL2THRESHUPPERCPU

Overload Level 3

Use the following configuration parameters for overload level 3 (see Chapter 5, "Provisioning the Cisco HSI"):

- OVLDLEVEL3PERCENT
- OVLDLEVEL3FILTER
- OVLDLEVEL3THRESHLOWERCALLS
- OVLDLEVEL3THRESHUPPERCALLS
- OVLDLEVEL3THRESHLOWERCPU
- OVLDLEVEL3THRESHUPPERCPU

Setting Overload Data

The following MML commands set overload data:

set-overload:level1|level2|level3:cpu, lower=*number***, upper=***number*

set-overload:level1|level2|level3:calls, lower=*number***, upper=***number*

set-overload:level1|level2|level3:gap, filter=normal|all, percent=*number*

The upper parameter specifies the threshold for overload detection, and the lower parameter specifies the hysteresis point at which the overload condition is removed.

The lower value should be greater than the upper value of a lower severity level.

For example:

set-overload:level1:cpu, lower=45, upper=50

set-overload:level1:gap, filter=normal, percent=50

set-overload:level2:cpu, lower=63, upper=70

set-overload:level2:gap, filter=normal, percent=75

set-overload:level3:cpu, lower=81, upper=90

set-overload:level3:gap, filter=normal, percent=95

These values would mean that:

- At less than 50 percent CPU usage, no call is gapped.
- From 50 percent to 70 percent CPU usage, 50 percent of calls are gapped.
- From 70 percent to 90 percent CPU usage, 75 percent of calls are gapped.
- At more than 90 percent CPU usage, 95 percent of calls are gapped.
- Before the overload level returns from level 3 to level 2, the CPU usage must fall to less than 81 percent.

Retrieving Overload Data

Use the **rtrv-overload** MML command to display the overload status and related overload data. For information about this command, see Appendix A, "MML Commands."

Logging

The logging level of one or more service packages is set using the **set-log** MML command. For more information about this command, see Appendix A, "MML Commands."

Rotating Log Files

Log files are rotated at system startup or when either of the following conditions occurs:

- The size limit for the corresponding file is reached. The size of the corresponding log file is equal to or greater than the value that the LOGFILEROTATESIZE configuration parameter specifies. The default value for this parameter is 10 Mb (see Chapter 5, "Provisioning the Cisco HSI").
- The age limit for the corresponding file is reached. The corresponding log file is equal to or older than the interval that the LOGFILEROTATEINTERVAL parameter specifies. The default value for this parameter is 1440 minutes (24 hours). See Chapter 5, "Provisioning the Cisco HSI," for more information about this parameter.

Log FileNaming Convention

Log rotation occurs when the system ceases to write to the current log file and commences to write to a new log file. The LOGFILENAMEPREFIX parameter defines the name of the active log file (see Chapter 5, "Provisioning the Cisco HSI"). The default is platform.log.

When log rotation is triggered, the existing file (for example, platform.log) is renamed with the format *platform_yyyymmddhhmmss.log* (see Table 4-4). For example, a platform error file rotated on September 30, 1999 at 12:36:24 is renamed platform_19990930123624.

Table 4-4 Log Filename Format

Format	Definition
LOGFILENAMEPREFIX	Provisioned filename (default is platform.log)
yyyy	Year
mm	Month
dd	Day
hh	Hour
mm	Minute
ss	Second

Note The time stamp is the coordinated universal time (CUT) from the machine at the time of rotation.

Log File Location

The LOGDIRECTORY parameter defines the directory for active log files and rotated log files (see Chapter 5, "Provisioning the Cisco HSI"). The default is $GWHOME/var/log/.

Log Messages

Log messages have the following format:

Date and timestamp, Package Name, <log level>, LogID:<text of the message>.

The following are examples of log messages:

```
Thu Dec  7 03:55:32:837 2000, Infrastructure, <DEBUG>, 205: GWModule Registration - 
shutdownList() - NbOfItems 10 - Item 8
Thu Dec  7 03:55:32:837 2000, Infrastructure, <DEBUG>, 206 : GWModuleRegistration - 
shutdownList() - NbOfItems 10 - Item 9
Thu Dec  7 03:55:32:838 2000, Infrastructure, <DEBUG>, 207 : GWReactor::thdId() returns 6.
Thu Dec  7 03:55:32:838 2000, Infrastructure, <DEBUG>, 208 : GWReactorModule::shutdown() - 
Thread has joined.
```

Log Message Packages

The following service packages can log messages:

- Application
- CallControl
- Connection
- DataManager
- Eisup
- FaultManager
- Gapping
- H323

- Infrastructure
- Overload
- ProcessManager
- Provisioning
- Signal
- Snmp
- SnmpSubagent
- Statistics
- Trace
- UserInterface

Logging Levels

Logging levels determine how much debug information is stored in the platform.log file for each package. Levels are set through use of a hexadecimal number between 0x0000 and 0xFFFF. 0x0000 is the lowest level, and will switch off logging for a particular package. 0xFFFF is the highest logging level.

Note We strongly recommend that you set all packages to log level 0x0000 in a live network. Set them to higher levels only when you debug on an offline network.

Setting Logging Levels

The **set-log** MML command dynamically alters the log level setting during the execution of the system. However, the **set-log** MML command does not affect the logging level of any current MML processes. For more information about the **set-log** command, see Appendix A, "MML Commands."

Note The enabling of logging will severely impact the performance of the HSI. We recommend that the HSI should be running at less than 2 calls per second when you enable logging. Logging will be automatically disabled when the HSI enters overload level 3. You can re-enable logging when the HSI exits overload.

RADVision Logging

The Cisco HSI application provides the capability (through MML) to initiate RADVision logging. The contents of the resultant log file are not under the control of the Cisco HSI application.

Use the **radlog** MML command to start and stop RADVision logging. RADVision logging can be directed to a file or into the standard logging output. For information about this command, see Appendix A, "MML Commands."

Gapping

The gapping level can be set from 0 to 100 percent. From 0 to 99 percent, the call type (normal or priority) is checked against the gapping level call status type. At 100 percent gapping, all calls are gapped regardless of call type.

Setting Gapping

To activate call gapping, complete the following steps:

Step 1 Determine the direction of the call to be gapped:

- Incoming (inc) for calls originating from the H.323 network
- Outgoing (otg) for calls originating from the PSTN Gateway (PGW 2200)
- Both (both) for calls originating from either side

Step 2 Determine what type of calls are to be gapped:

- Normal calls (nonpriority calls)
- All calls

Step 3 Determine the percentage of calls to be gapped. The percentage can range from 0 to 100 percent. If 100 percent is selected, all calls are gapped, regardless of the type of call.

Step 4 Enter the **set-gapping** MML command. For example, to gap 60 percent of all calls for both directions, enter:

set-gapping:both:calltype=all,percent=60

Retrieving Call Gapping Data

To retrieve the current levels of call gapping for all gapping clients, enter the **rtrv-gapping** command. The following text displays:

```
Client Name      Direction        Level            Call Type        Active

Overload         Outgoing         10               Normal           No
Overload         Incoming         10               Normal           No
MML              Outgoing         20               All              Yes
MML              Incoming         30               All              Yes
```

The output shows the gapping levels set by the overload function and the MML command **set-gapping**. The highest gapping level is used as the level to gap calls, which is indicated as Yes in the column titled Active. In this example, the MML levels for outgoing and incoming calls are active.